ISSUE #5

MOSAIC™ 64K SELECT NEWSLETTER

64K COMPATIBILITY

One of our customers biggest concerns, and rightly so, is compatibility. Though it is not financially nor reasonably possible to test every piece of software on the market, we try to stay on top of any problems that our customers may encounter.

When the Atari computer was first introduced a few years ago, Atari stated very plainly that the maximum amount of RAM the computer can access was 48K. Thanks to Free Enterprise and the American Way (and Wynn Smith's ingenuity) this did not remain the case for very long. Software companies, on the other hand, took Atari's word to heart and wrote their programs accordingly. When a company writes a piece of software that requires a minimum amount of RAM (say 48K) to run, the software must check to see if the computer has the required amount of RAM before it executes. The software does this by comparing RAMTOP, address 106 decimal or 6A hex, with a number the author determines is valid. The value stored in RAMTOP is the number of 256 byte pages of free RAM available. If the software requires 48K, then the number in RAMTOP would need to be 192 decimal or C0 hex. This number is achieved by dividing the amount of required RAM (49152) by 256 equaling the number of free pages. Sounds simple because it is. The problem comes when the author assumes that there will never be more than 48K and RAMTOP can never be greater than 192 decimal or C0 hex.

Then came Mosaic with the 64K Select. When the 64K Select powers up (without a cartridge) the computer sees a total of 52K of free RAM and RAMTOP reflects the change by displaying 208 decimal or CF hex. If the author wrote his RAMTOP compare statement such as; IF PEEK (106)<> 192 THEN END, we're in trouble because this statement (and its machine language counterpart) are looking for a specific answer. Even though we have more than enough memory to accomodate the program, it will abort. If the author had written the statement like; IF PEEK (106) >= 192 THEN GOTO EXECUTE, all would have been well because the statement looked for a minimum but not a maximum. If you encounter a program that won't load because of a "not enough memory error" or "REMOVE CARTRIDGE AND REBOOT" when no cartridge is installed, this is probably the reason why.

This problem is easily fixed for the owner of a 400/800 64K Select by using the software supplied with the owners manual called 48K Boot Fix. Unfortunately, the 400 ONLY 64K Select owners fix is not quite so easy. The Nov-Dec newsletter describes the fix in detail with the <u>exception of step one</u> (I goofed, sorry). Step one should read: Locate pin 11 of U17, cut the trace that extends from pin 11 to the right. Make sure you have located the right trace.

If for some reason you didn't get te 48K Boot Fix with your owners manual, please let us know. We will be happy to send it to you.

M: PART 2

Considering all of the comments that we received on last month's article, it is clear that we should have explained the goals and purposes of the memory manager in more detail. We thank you for your comments and suggestions.

The main goal of the memory manager is to create a complete environment for the programmer in which all address specific references and all concern for memory allocation conflicts can be eliminated. The degree to which this goal can be accomplished is dependent upon the 6502 microprocessor and the Atari operating system.

With the main goal in mind we have divided our memory manager into three parts. The first part is called M: Traditional. It's purpose is to allow assembly language programmers to access data in a manner that is consistant with the Atari Operating System. Most of this article will focus on this part of M:.

The second part of the memory manager is called M: Variable. It's purpose is to allow an assembly language programmer to create data spaces and to directly access that data space with instructions such as LDA and STA with the exact location of the data space is controlled by the memory manager. The advantage of M: Variable is that the programmer can create data spaces without concern for creating memory conflicts. He can access data directly in order to eliminate the need for data buffers and to speed program execution since data will no longer be required to physically move from buffer to buffer.

The third part of the memory manager is called M: Task. It's purpose is to allow a programmer to develop software that is not address specific and to pass control from one program module to another without knowing the exact location of that module.

One advantage of M: Task is that software development becomes easier. Program modules can be developed independently and addressed as complete entities rather than pieces of a bigger puzzle. Another advantage is that Virtual Memory techniques can be developed so that the application program and it's data spaces can be much larger than physical RAM. However, the most important advantage of the memory manager is that multiple programs can be loaded and executed simultaneously.

The details of M: Variable and M: Task will be explained in future articles.

The data tables that were described in last month's article were for the use of M: Traditional. Thanks to the suggestions that we have received, those tables have changed somewhat. From now on we will refrain from publishing specific

details until the major ideas have been presented. This will allow us to receive more suggestions without wasting your time with revised updates.

The main purpose of M: Traditional is to provide a glorified disk emulator and to provide a solid foundation for the development of M: Task and M: Variable.

All DOS commands except Format are supported, including Open, Close, Putchr, Getchr, Putrec, Getrec, Status, Rename, Delect, Lock, Unlock, Note, and Point. Some new commands will be added.
Secret stops the file name from being listed in a directory read operation.
Show removes the Secret status.
Bind binds a file to its existing memory location so that it will not be moved during "house-keeping" operations.
Unbind removes the Bind status.

The Note and Point commands will be compatible with DOS 3. That is, the file pointer works by specifiying a relative byte count from the beginning of the file.

All file communication at the M: Traditional level is through an IOCB to a named file. All data is passed through a user specified buffer. All command numbers are the same as those for Atari DOS. The command numbers for Secret, Show, Bind and Unbind have not been determined.

The most obvious advantage that M: Traditional has over disk emulation is that it can be implemented while maintaining all four disk drives. The most important advantage is that direct address files can occupy bank memory while the memory manager fills empty spaces with user data. This will allow the speed advantages of disk emulation while eliminating memory conflicts wih anything that must occupy bank memory. All existing software can be easily changed to use M:.

The memory manager is an ambitious project. Many details have yet to be worked out. We would like to hear from you. Please write if you have suggestions, comments, or questions concerning any part of M:.

M: Mosaic Electronics
P. O. Box 708
Oregon City, OR 97045

Next month we will discuss part two of the memory manager, M: Variable.

The 128K Atari 400

As we all know, Atari computers rate among the best personal computers on the market. Their advanced operating system and superior graphics capibility have long been the envy of other manufactures. With all their assets they still had limited RAM capacity. We solved this problem over a year ago, with the advent of our 64K RAM SELECT. The RAM SELECT allowed the 800 user up to 192K, far beyond the wildest expectations of the original engineers. On the other hand the 400 user was limited to 64K, but not necessarily. Electronically, the 400 and 800 are almost the same, and for our purposes we may consider them to be so, with the major internal physical differences being the lack of any expansion slots. If we are to expand the 400 to 128K we will have to add another expansion slot or pseudo slot. Since this project requires about $600 in equipment and some good technical know-how, it will take a reasonable amount of intestinal fortitude to undertake. Let me warn, <u>if you don't know what you're doing, don't do it</u>. **SEE NOTE** Mosaic will do the upgrade for $25 and is offering the 400/800 RAM SELECT board for $149.95 thru February not including shipping. For those of you who have the expertise and will, this article will allow you to increase your 400's RAM size far beyond what was once thought possible.

Equipment needed includes: 25-30 watt pencil tip soldering iron, hand drill and bits, good quality fine guage resin core solder, two sections of 25 conductor ribbon cable about 12 in. long, electricians tape, two plasic bags approximatly the same size as the ram boards, skill, patience.

This modification can only be done with the 400/800 RAM SELECT board. You can verify this is the board you have by looking on the <u>back of your owners manual</u>. The series number should be L-229D. If you do not have the correct board you may call us and ask about our exchange program.

Locate the board diagram on the bottom of page 8 of your owners manual. This diagram shows the various shorting strap locations on the ram board. These straps must be set according to configuration 7 on page 15. Before we go any further set the straps on each board.

Since we need two ram sockets and we only have one we are going to use the ribbon cable to <u>parallel</u> the card-edge connectors of the two ram boards. If you are not sure what paralleling is <u>STOP HERE</u>. All pins must be tied together electrically through the ribbon cable. Pin one to pin one, pin two to pin two, etc., etc.. The ribbon cable will be soldered to the <u>top</u> of the card-edge connectors on both boards so that it won't interfere with the ram card being plugged back into the ram socket on the mother board. The cable must be arranged so that the two ram boards may be placed parallel to one another. One ram board will be inserted into the ram socket and the other will float between the socketed ram board and the CPU card (see figure one.) This configuration is certainly not ideal but

is necessary due to the limited space of the 400. Cut the ribbon cable diagonally from the end so that you may run it 90 degrees with respect to the card-edge connector (see fig. two.) All 44 pins must be tied together through the two pieces of ribbon cable. After all soldering is complete and you're confident all connections are sound, try placing the two boards against each other as in figure one. Adjust the cables in such a way that they are not streched or binding.

Locate the two rows of gold pins at the top right of each ram board. These pins are signal lines from the mother board that are not available through card-edge connector. The bottom set of pins on the board that will go into the ram slot will be connected to the cables of the mother board. Using another piece of ribbon cable, parallel the upper pins of both boards.

Now that these steps have been completed and all looks good ,check all solder joints making sure you haven't crossed any of the wires. Make sure the ribbon cable lays flush between the boards and is not going to interfere with installation.

Insulate the floating board from touching anything. Apply electricians tape to the card-edge connector and put the plastic bags around the board in any configuration that will not allow any of the boards to touch. This requires much care, if the boards touch, they're gone.

Install the two boards onto the mother board being very careful not to let any of the boards touch. Before replacing the mother board into the metal shield, drill about ten 3/8 to 1/2 inch holes in the aluminum housing for added ventilation. Reassemble the computer and test, if any problems are noted check solder connections and possible shorts.

Summary

1. Set jumpers according to page 15 figure 7
2. Solder ribbon cables to card-edge connector (44 pins) in parallel
3. Solder ribbon cable to gold pins in upper right hand corner of ram boards in parallel.
4. Check all solder connections (i.e. cold joints, shorts, messy)
5. Insulate the boards from one another with plastic and electrical tape.
6. Drill approx. 10 holes in the aluminum shield for added ventilation. (this is necessary)

*** NOTE: Due to the varried skill levels involved, we are not able to extend our gaurantee to cover this procedure. If Mosaic performs the modification, all guarantees will remain intact.

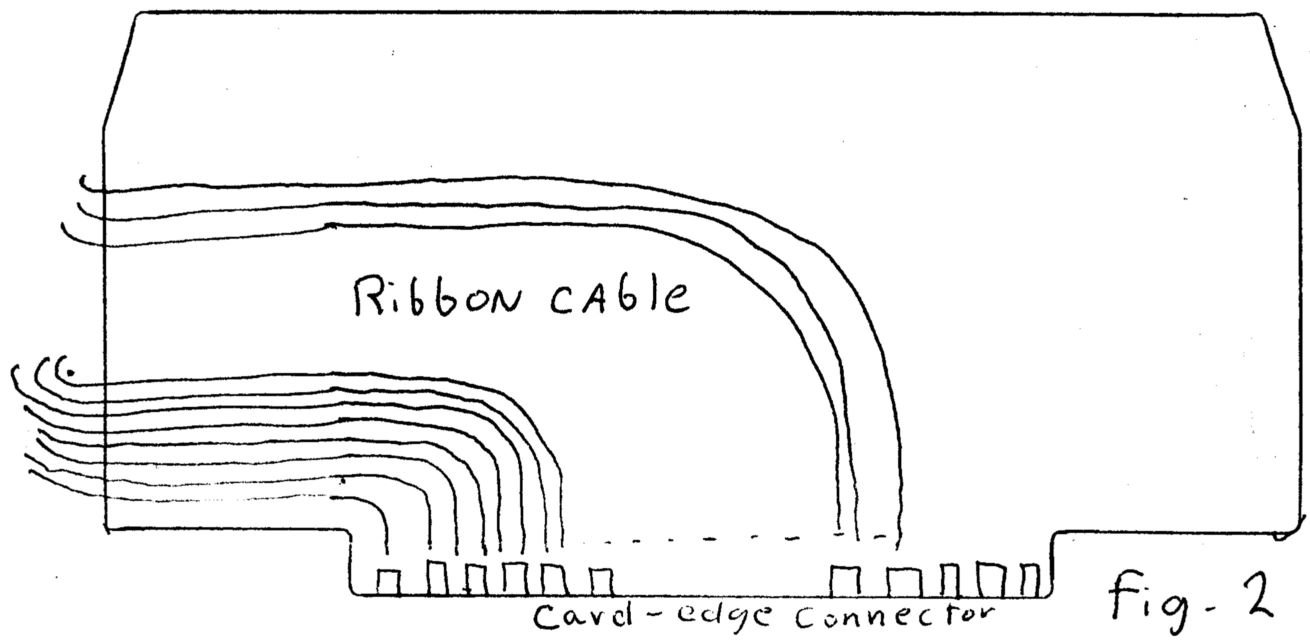
Ribbon CAble
Card-edge connector

Fig. 2

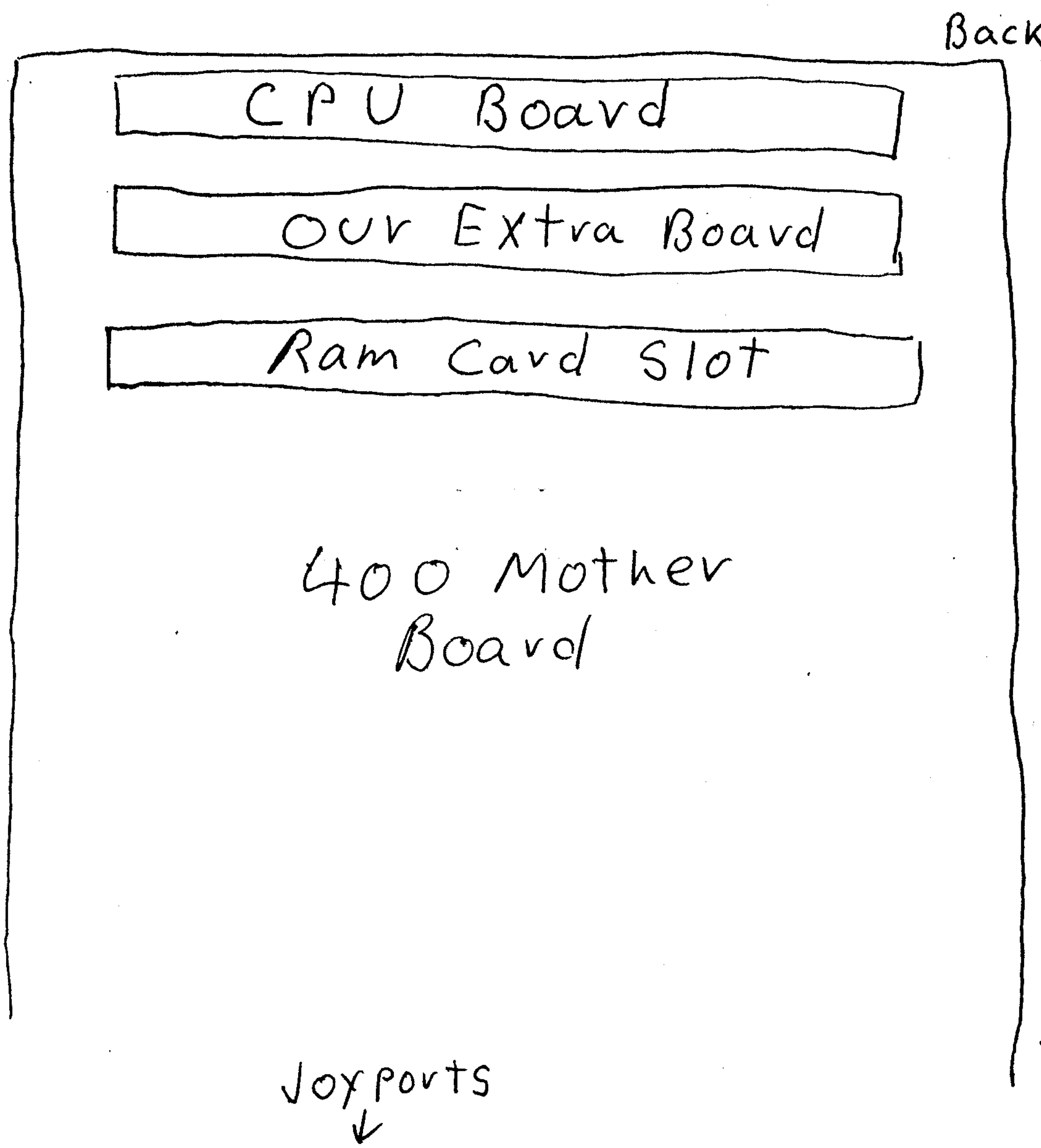
Back
CPU Board
Our Extra Board
Ram Card Slot
400 Mother
Board
Joyports

Fig 1

BULK RATE
U.S. POSTAGE
PAID
Permit No. [illegible]
[illegible] City, OR

[redacted]